



WAVA: a New Web Service for Automatic Video Data Flow Adaptation in Heterogeneous Collaborative Environments

Jean-Baptiste Aupet, Rami Kassab, Jean-Christophe Lapayre

► To cite this version:

Jean-Baptiste Aupet, Rami Kassab, Jean-Christophe Lapayre. WAVA: a New Web Service for Automatic Video Data Flow Adaptation in Heterogeneous Collaborative Environments. CDVE'09: 6th Int. Conf. on Cooperative Design, Visualization and Engineering, 2009, Luxembourg, Luxembourg. pp.280–288. hal-00563302

HAL Id: hal-00563302

<https://hal.science/hal-00563302>

Submitted on 4 Feb 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

WAVA: a New Web Service for Automatic Video Data Flow Adaptation in Heterogeneous Collaborative Environments

J.-B. Aupet, R. Kassab, and J.-C. Lapayre

LIFC - EA 4269, Université de Franche-Comté,
16, Rte de Gray - 25030 BESANCON Cedex, France.
[jbaupet,rkassab,jclapayre]@lifc.univ-fcomte.fr
Web page: <http://lifc.univ-fcomte.fr>

Abstract. The progressive needs of video streaming for different applications in varied domains have created a new set of heterogeneous environments especially in Virtual Collaborative Environments. In order to get the best performance of such environments, video streaming has to be adapted to the different parameters that characterize these environments, namely: bandwidth, CPU, GPU, screen resolution, etc. In this paper, we define a new Web Service, named Wava (*Web Service for Automatic Video Data Flows Adaptation*). Wava allows multimedia platforms and collaborative application servers to adjust the adaptation at two levels: at the static level during initialization and at the dynamic level according to the variation of the environment.

Keywords : Adaptation, Asynchronous, Collaborative Environments, Multimedia, Real-time, Streaming, Synchronous, Terminal, Video, Web service.

1 Introduction

Multimedia environments providing access to media contents from heterogeneous environments are commonly called UMA "Universal Multimedia Access". In heterogeneous environments systems clients differ in their hardware and software capabilities and limitations. These differences might be found on the communication level, for example the variation of bandwidth from 50 kb to 20 Mb. Variations may also involve the terminal specifications *e.g.* CPU, memory, display capabilities and GPU encoding-decoding capacities. Therefore when we talk about video adaptation systems we should take all conditions into account. By heterogeneous applications we mean the variation of usage, entertainment or work (for example tele-medicine) and programming languages. These applications are distinguished between two types: synchronous and asynchronous applications. In order to solve these differences, and because we mostly work on the Internet, XML files and Web Services are involved.

In the first section of this paper we present the state of the art in this domain. In the following section, we define our new Web Service Wava (*Web Service for*

Automatic Video Adaptation) which allows platforms to adapt the stream at two levels: at the static level (during initialization) and at the dynamic level according to the variation of the environment, and its performance. The last section presents the conclusion and future work.

2 State of the Art

The development of multimedia applications puts in evidence the necessity of new research. The arrival of HD video by the Internet and the increase of multiple formats of video generate a lot of studies in each part of this domain. In this section, we focus our study on adaptative systems and especially on UMA (Universal Media Access) systems based (Wang 2007).

2.1 Adaptation with the UMA Model

The General architecture of the UMA system consists of four main components described in (Wang 2007) and illustrated in the figure 1:

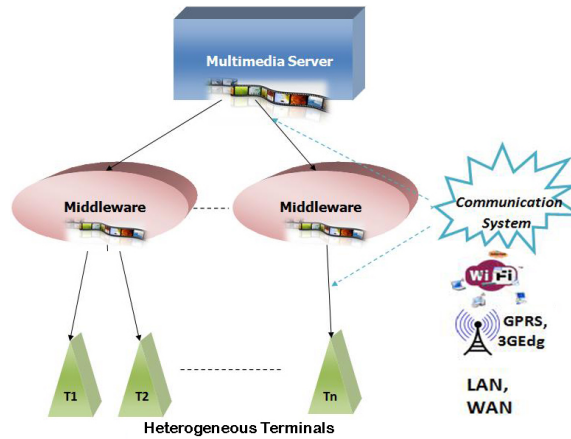


Fig. 1. UMA Adaptation System

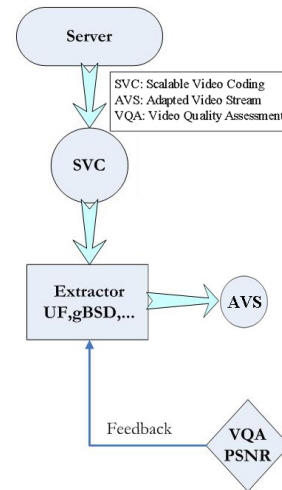


Fig. 2. System Chart

Multimedia server

Research which focuses on the adaptation aspect performed by the multimedia server (Prehofer 1999) considers the server is responsible of coding and forming the high quality Scalable Video bitstream (Sprljan 2005). That means that it

can be scaled (adapted) to be in adequacy with specifications. In addition, the server can adapt the video. However, the adaptation process consumes larger amount of calculation power than a real time system can actually bear. In order to further adapt the video bitstream in a convenient way, the middleware has to decode and then to recode the bitstream to the adapted format. Therefore, the server transmits specific bitstream description helping the middleware to avoid the decoding-recoding step. The server role in adaptation system can be summarized to, forming scalable bitstream (Sprljan 2005), performing adaptation, and transmitting video information with the SVC (figure 2).

Middleware.

The main purpose of middleware is the adaptation process, it receives the scalable bitstream, operates the adaptation and then sends the scaled bitstream to the terminal. In real time systems, studies are concentrated on middleware and proposed adaptation solutions like UF (*Utility Function*) (Chang 2002), (Kim 2005), (Wang 2007) or *gBSD* (generic Bitstream Synt. Desc.) (Iqbal 2007). The main factor used in the adaptation process is the constraints (Feedback) received from the terminal. The feedback consists of terminal-transmitted information necessary for helping the middleware in the adaptation process to choose the right video bitstream.

Communication system

The revolution of communication systems in the last decade has created various media, paved the way for real-time Collaborative systems. From this viewpoint, it is very important to analyze the communication mediums for the adaptation. Terminals can use a medium to communicate and pass to an other. This step is the main reason for performing the adaptation. For example, a mobile client could change his Internet connection from GPRS to Wi-Fi (Ruiz 2002). If the system allows a client to notify the system of these changes, it is called an adaptable system. If the system is able to notify itself about the changes of the environment, it is called an adaptive system.

Terminal

In addition to decoding the video, terminals have one of the most important tasks in adaptation systems that is creating feedback. Terminal application has to calculate the feedback and send it to the middleware or the server, which in turn should make the right decision. Many factors can influence the feedback, such as computation capacity of the terminal or free resources to store video and others. The most important resources to take into account are: Bandwidth and its variation, the available RAM and Hard disk capacity to store buffered video, the CPU capacity and load, and GPU Capacity for Displayable Resolution and codecs supported Software.

Feedback is the major factor to choose the adapted video, many techniques are used to calculate it: produced as a mark from the terminal, created either by testing the hardware of a terminal or by using mathematical methods such as PSNR (Wang 2004) or VQA (Zhai 2008) (see part 2.4).

2.2 Use of adaptation in video applications

Collaborative environments are used more and more due to the increase of data transmission capacity. Now a user can easily be noticed by chat or voice-chat, and more and more by video-conferencing. Video applications vary from entertainment purposes like watching films on VOD and channels of television but also for medical applications, such as video conferencing for Telediagnosis(Jung 2005), e-learning. . . These applications are classified in two main categories, real time (synchronous) video applications and stored video (asynchronous) applications.

Stored video applications These kind of applications are easier to perform, there is no need to consider the time factor in all parts of the system. A video is available on the server and the application has to adapt it with terminal capabilities.

Real time applications These applications need special solutions to ensure the time contract and quality of service are respected. Collaborative applications must ensure synchronous data exchange and consistency. The obvious the goal of the middleware is: saving time for the adaptation process.

2.3 Multidimensional video adaptation

In terms of performance, the adaptation process has three main factors or three main dimensions which are: time, resolution, and quality. And they are referred as: temporal, spatial and SNR, respectively:

- Temporal (Kropfberger 2004) and (Leopold 2003): Frame rate (from 1 up to 30 fps fig. 3a.)
- Spatial: Screen and GPU Resolution capabilities (figure 3b.)
- SNR: Signal-to-Noise Ratio (Low, mid, high): Extended for video by PSNR and VQA in depth.

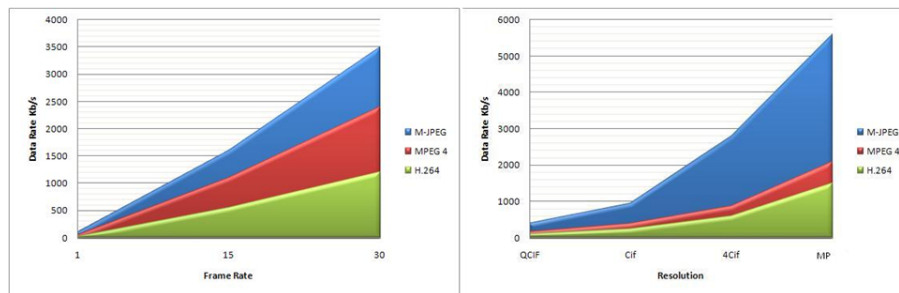


Fig. 3. Bandwidth consumption depending on: a. Frame rate and b. Resolution

2.4 PSNR and VQA in depth

PSNR: Peak Signal to Noise Ratio is a method to calculate the noise ratio of a signal depending on the measurement of MSE (Mean Squared error) (Wang 2004). However, PSNR does not compute the real performance of a video stream. For this reason, it is not applicable to measure the performance of the multidimensional video adaptation process. Furthermore, it has been proven that there can be different videos with different qualities that have the same PSNR. These experiments were correlated by HVS (Human Visual System) in (Zhai 2008), which depends on human notice for giving the result. HVS is also called perceptually VQA.

VQA: Video Quality Assessment is a value that is calculated in subjective (HVS) or objective (mathematical) manner (Zhai 2008). In this paper, we focus on the objective method in calculating VQA, because it can be calculated mathematically. The differences between these VQA types are the amount of video features sent with the adapted video.

3 A New Web Service for Automatic Video Adaptation: WAVA

When users have to communicate, they do not have to be impacted by the weaknesses of the others. Clients do not feel the latency of the adaptation process. But they can not determine themselves the optimal feedback all the time. With regard to this, we propose a solution that automatically and periodically detects the changes of the environment without user's interaction. In order to provide transparent processing, enhance the performance and facilitate the usage of the system, we propose the following architecture.

3.1 WAVA Architecture

In the beginning of our work, we used a client-server JAVA application, where the client sends his mark manually and the server reacts according to this mark. We automatized this process by making a simple detection of the load of the CPU and the network congestion. In another experience, we created a client application that reads the terminal performances (CPU frequency and load, Screen resolution, network bandwidth) and sends the corresponding mark to the server. The latency generated by the calculation did not allow an efficient and satisfactory work. Furthermore in a collaborative environments, heterogeneous client can connect to the system. The best solution to resolve this problem is the use of Web Services. Because of their portability and independence from programming languages, Web Services can handle any kind of platforms. Furthermore, we save the feedback calculation time of the terminal. When the client chooses a video, the collaborative system automatically detects terminal performances with the WAVA marks and sends back adapted videos.

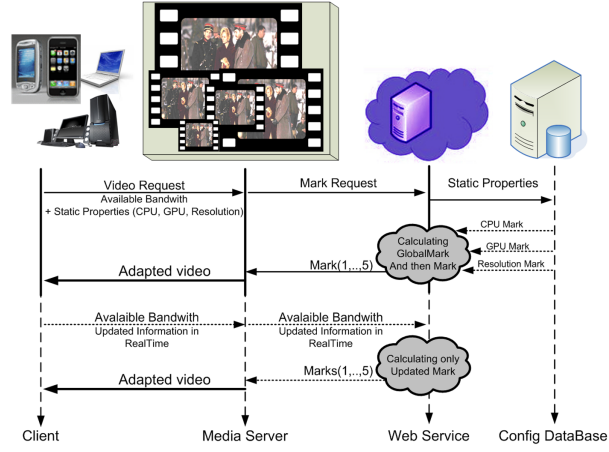


Fig. 4. WAVA Processing

So we propose a new Web Service design that detects each terminal property and calculates the related mark. Depending on this mark, the server decides whether it sends a high or low quality video matching the terminal's exigences. The quality is determined by the decrease of the resolution and frame rate. It is possible to integrate this system in a web page with a script that reads the terminal performance and sends an XML file in a Virtual Awareness Card format like in (Aupet 2009) to the WAVA web service. Subsequently WAVA uses an algorithm to produce the corresponding mark. At the media server we have two choices. The first is to make a video pool that contains all the videos in all the codec formats, which match with all the marks: the same video will be repeated for all formats. The second choice is to use one video source that would be converted in real-time either by the server or by the middleware to adapt it to the marks received from the Web Service. We have chosen the second solution for the video conferences, where we have the camera as the only source. So we directly adapt the stream to match the feedback.

3.2 WAVA Processing

The *Mark* represents the terminal performance. We classify CPUs, graphic cards, resolutions, and bandwidth into groups. Each group contains the individuals that perform in an identical manner in terms of video streaming. Then, a mark is attached to each group. We apply this design by using a database which stores devices attaching to their marks. The database is automatically filled with the unknown devices of terminal by giving a mark calculated with the average mark of the terminal performance. We distinguish two categories of mark: the static mark that does not change during a session, we call them Global Marks (Mg).

This mark refers to CPU (Mcpu), resolution (Mres), and graphic card (Mgrph), and obviously remain unchangeable during one session. Mg is calculated as the minimum value of Mcpu, Mres and Mgrph. The second category of mark is the dynamic mark that refers to the bandwidth (Mbw) that changes during the video streaming session. In order to get the final mark (M), we calculate the minimum value between Mg and Mbw.

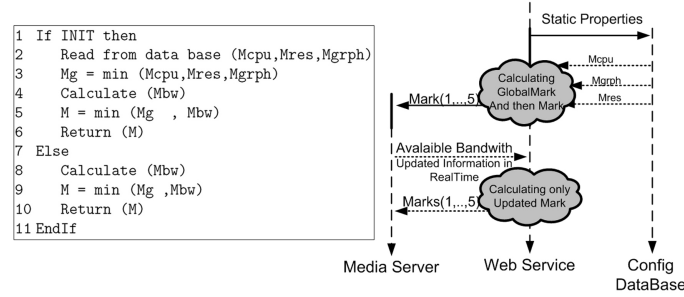


Fig. 5. Wava Algorithm

We always use the smallest value because we need the system work optimally: we adapt our collaborative system to the weakest value. After calculating the mark, WAVA sends it and saves the global mark. Periodically WAVA receives the bandwidth information of users, calculate the new Mbw, then calculates the mark which remains the minimum between the Mg and the updated Mbw.

3.3 WAVA Performance

In order to test our system, we make a pool of videos in different formats to test several terminals. With these videos we have tested the proportion of packet loss during the streaming transfer. We used the Mpeg-H.264 codec to produce these videos in five different qualities. We changed the resolution and frame rate. Results are in the figure 6a.

Periodically the bandwidth mark is sent and the new adaptation mark is updated to provide the best video quality that fits the real conditions of the user. We describe a scenario-based test (fig. 7) to show the decrease of packets loss by using WAVA Web Service. We can show the technical performance of WAVA in figure 6b. In facts, when the adaptation is well done, participants can work efficiently.

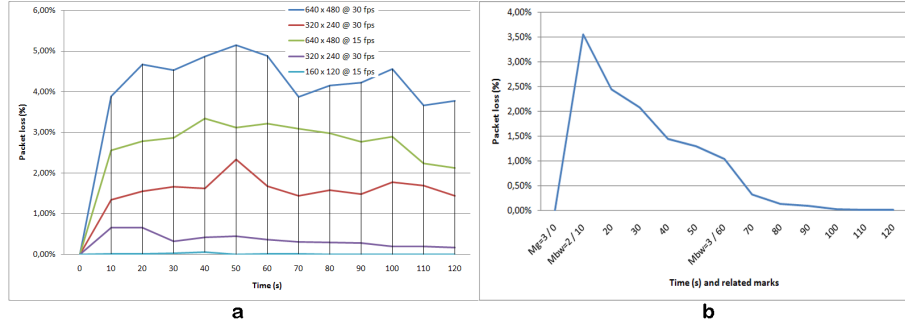


Fig. 6. a.Packets loss - b.Wava Performance

```

Init :
    CPU == P3 && RES == 4CIF && GPU == GeForce256
    Mcpu = 4 ; Mres = 3 ; Mgpu = 3 ;
    Mglobal = min(Mcpu,Mres,Mgpu);
    SendMark (Mglobal); //Adaptation with Mark=3
OnStart : DetectBandwith(BW);
    BW == 512 Kb
    Mbw = 2 ; Mark = min (Mglobal , Mbw);
    SendMark (Mark); //Adaptation with Mark=2
every min : DetectBandwith(BW);
    BW == 2 Mb
    Mbw = 3;
    Mark = min (Mg , Mb);
    SendMark (Mark); //Adaptation with Mark=3
End

```

Fig. 7. Scenario-based Test

4 Conclusion and future work

In this paper, we showed that video adaptation is one of the main matters of multimedia development in heterogeneous collaborative environments. Therefore researche has been conducted in this domain which focused on the UMA model. We proposed a new Web Service based system for automatic video adaptation (WAVA). The Web Service is used to calculate the best adaptation that matches the system capabilities, for both, static parameters (during initialization) and dynamic parameters depending on the variation of the environment. We implemented and tested our system, and we exposed the performance results. Wava significantly improves the use of mobile terminals for collaborative work.

Our future work focuses on how to test Wava not only with a mark for each terminal, but also for a whole collaborative environment. We imagine a process to adapt media during a session of medical tele-diagnosis between remote mobile practitioners: we are working on new distributed algorithms for shared marks calculation and optimising adaptation to each participant.

5 Acknowledgements

The authors thank the European Community (European FEDER) for financing this work by the InterRegIV project SERVASTIC : Teleneurology in Mobile Environments.

References

- (Aupet 2009) J.-B. Aupet, N. Elmarzouqi, E. Garcia, and J.-C. Lapayre, Virtual Awareness Card for adaptability in Collaborative Virtual Environments, *In SETIT 2009, 5th Int. Conf. on Sciences of Electronic, Technologies of Information and Telecommunications*, pp.243, Mar.
- (Chang 2002) Shih-Fu Chang, Optimal Video Adaptation and Skimming Using a Utility-Based Framework, *Proc. IWDC-2002*, pp.20, Capri Island, Italy.
- (Iqbal 2006) Razib Iqbal, Shervin Shirmohammadi and Chris Joslin, MPEG-21 Based Temporal Adaptation of Live H.264 Video, *8th IEEE ISM'06*, pp.457-464, Dec.
- (Iqbal 2007) Razib Iqbal, Shervin Shirmohammadi, and Abdulmotaleb El Saddik, A Framework for MPEG-21 DIA Based Adaptation and Perceptual Encryption of H.264 Video, *Multimedia computing and networking*, vol.6504(03), pp.1-12, (2007).
- (Jung 2005) Chulho Jung, Sanghee Lee, and Eunseok Lee, Dynamic Adaptive Architecture for Self-adaptation in VideoConferencing System, *School of Information and Communication Engineering, LNCS 3768, Springer-Verlag*, pp.36-47, Heidelberg.
- (Kim 2005) Jae-Gon Kim, Yong Wang, Shih-Fu Chang, Hyung-Myung Kim, An optimal framework of video adaptation and its application to rate adaptation transcoding, *ETRI Journal*, vol.27(4), pp.341-354, Aug.
- (Kropfberger 2004) Michael Kropfberger, Klaus Leopold, and Hermann Hellwagner, Quality Variations of Different Priority-based Temporal Video Adaptation Algorithms, *IEEE Workshop on Multimedia Signal Processing No6*, pp.183-186, Sept.
- (Leopold 2003) Klaus Leopold, Hermann Hellwagner, Michael Kropfbereer, Quality Controlled Temporal Video Adaptation, *SPIE proceedings series Congress Internet multimedia management systems IV*, Vol. 5242, pp.163-174, Sept.
- (Prehofer 1999) Christian Prehofer, Wolfgang Kellerer, Robert Hirschfeld, Hendrik Berndt, Katsuya Kawamura, An Architecture Supporting Adaptation and Evolution in Fourth Generation Mobile Communication Systems, *Journal of Communications and Networks*, vol.4, pp.336-343, Jan.
- (Ruiz 2002) Pedro M. Ruiz, Emilio Garcia, Improving User-Perceived QoS in Mobile and Wireless IP Networks Using Real-Time Adaptative Multimedia Applications, *Proceedings of PIMRC2002*, pp.1467-1471, Sept.
- (Sprljan 2005) N. Sprljan, M. Mrak, G. C. K. Abhayaratne, E. Izquierdo, A Scalable Coding Framework for Efficient Video Adaptation. *6th WIAMIS*, Vol.1, Apr.
- (Wang 2004) Yong Wang, Shih-Fu Chang, Alexander C. Loui, Subjective Preference of Spatio-Temporal Rate in Video Adaptation Using Multi-Dimensional Scalable Coding, *IEEE ICME*, vol.3, p.1719-1722, Jun.
- (Wang 2007) Yong Wang, Jae-Gon Kim, Shih-Fu Chang, and Hyung-Myung Kim, Utility-Based Video Adaptation for Universal Multimedia Access (UMA) and Content-Based Utility Function Prediction for Real-Time Video Transcoding, *IEEE Transactions on Multimedia*, vol.9(2), pp.213-220, Feb.
- (Zhai 2008) Guangtao Zhai, Jianfei Cai, Weisi Lin, Xiaokang Yang, Wenjun Zhang, Three Dimensional Scalable Video Adaptation via User-end Perceptual Quality Assessment, *IEEE Transactions on Broadcasting*, vol.54, pp.719-727, Sept.